

Dependências de projetos Python: requirements.txt

Jan 6, 2018 • 🕒 Reading time ~2 minutes

Author note: You can [read this post in English here \(https://jtemporal.com/requirements-txt-en/\)](https://jtemporal.com/requirements-txt-en/).

Depois de uma conversa longa outro dia sobre dependências e reprodutibilidade de ambientes, decidi fazer essa colinha pra falar de um arquivo muito frequente nos projetos Python (<http://python.org/>) que vejo por aí: o `requirements.txt`.

Um passo comum da configuração de ambientes de desenvolvimento Python é fazer a instalação de dependências. Muitas vezes esse passo é executado da seguinte forma:

```
$ pip install -r requirements.txt
```

Mas o que é de fato esse arquivo? Segundo a [documentação do pip sobre arquivos requirement \(https://pip.pypa.io/en/stable/user_guide/#requirements-files\)](https://pip.pypa.io/en/stable/user_guide/#requirements-files):

"Requirements files" are files containing a list of items to be installed using pip install

Ou seja, esse arquivo nada mais é do que um arquivo de texto, contendo uma lista de itens/pacotes para serem instalados durante o `pip install`.

Vamos a um exemplo

Considerando esse `requirements.txt` aqui:

```
1 -e git+https://github.com/jtemporal/caipyra.git@master#egg=caipyra  
  (https://github.com/jtemporal/caipyra.git@master#egg=caipyra).  
3 seaborn==0.8.1  
2 pandas>=0.18.1  
4 serenata-toolbox
```

Temos várias informações importantes, vamos repassá-las linha a linha:

1. **-e git+https://github.com/jtemporal/caipyra.git@master#egg=caipyra (https://github.com/jtemporal/caipyra.git@master#egg=caipyra)**: Dessa forma conseguimos instalar pacotes Python que estejam disponíveis no GitHub mas não no PyPI. Essa é uma dica muito legal quando por exemplo, você precisa da versão em desenvolvimento de um biblioteca ou quando você prefere usar um fork no lugar da versão tradicional do pacote.
2. **seaborn==0.8.1**: No caso do seaborn, estamos instalando a versão `0.8.1`. Fixar a versão dessa forma é interessante pois garante que o seu projeto vai sempre estar funcionando já que mudanças nos pacotes são indicadas pela alteração no número da versão.
3. **pandas>=0.18.1**: Da mesma forma que o sinal de `==` define uma versão específica a ser instalada, quando usamos o sinal de `>=` nessa lista estamos dizendo que queremos instalar qualquer versão da biblioteca, nesse caso o pandas, seja ela a versão `0.18.1` ou uma mais recente. Interessante nesse caso é notar que você pode definir um intervalo de versões, por exemplo, `pandas>=0.15.0, <=0.18.1`.
4. **serenata-toolbox**: Quando não especificamos uma versão, o pip sempre tentará instalar a versão mais recente do pacote especificado.

Criando seu requirements.txt

Um jeito simples de criar o seu próprio arquivo de depências é usando o comando `pip freeze` da seguinte forma:

```
$ pip freeze > requirements.txt
```

O comando `pip freeze` lista no terminal os pacotes Python instalados no seu ambiente já no formato que o pip install consegue entender, ao associá-lo com o operador de redireção `>` você consegue escrever a mesma lista que aparece no terminal dentro do arquivo de texto.

Massa né? Agora é só criar arquivos de dependências para todos projetos 🤪

Links

- [Caipyra \(https://github.com/jtemporal/caipyra\)](https://github.com/jtemporal/caipyra)
- [pandas \(https://pandas.pydata.org/\)](https://pandas.pydata.org/)
- [seaborn \(https://seaborn.pydata.org/\)](https://seaborn.pydata.org/)
- [serenata-toolbox \(https://github.com/datasciencebr/serenata-toolbox\)](https://github.com/datasciencebr/serenata-toolbox)
- Artigo em inglês sobre [operadores de controle e redireção \(https://unix.stackexchange.com/questions/159513/what-are-the-shells-control-and-redirectation-operators\)](https://unix.stackexchange.com/questions/159513/what-are-the-shells-control-and-redirectation-operators)

Agradecimentos

Obrigada aos amigos que participaram da conversa sobre dependências e reproducibilidade de ambientes <3

TAMBÉM EM JTEMPORAL

Projetos Brasileiros para fazer pull ...

2 anos atrás • 5 comentários

Versão 2019 da lista de projetos brasileiros para contribuir no #Hacktoberfest

Dependências de projetos Ruby: Gemfile

4 anos atrás • 1 comentário

Entenda o que é uma Gemfile e como criar uma caso não exista.

Copying files into a container

2 anos atrás • 1 comentário

Let's learn to copy files into a container without using volumes

Formatan usando o

4 anos atrás

Coisa linda um formata principalme

8 Comentários jtemporal  Disqus' Privacy Policy

 Entrar ▾

 Recomendar 8  Tweet  Compartilhar

Ordenar por Mais votados ▾

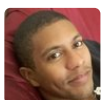


Participe da discussão...

FAZER LOGIN COM

OU REGISTRE-SE NO DISQUS 

Nome



Rodrigo Silva Barretos • 3 anos atrás

Post simples e esclarecedor o suficientes. Obrigado!

21 ^ | ▾ • Responder • Compartilhar >



Mural Do Marujo • 3 anos atrás

Valeu bródi, pelas dicas legais e esclarecedoras, em poucas palavras tu consegui sanar minha dúvida.

1 ^ | v • Responder • Compartilhar >



Jessica Ferreira • 10 meses atrás

Ficou simples de entender. Parabéns!
Muito obrigada!

^ | v • Responder • Compartilhar >



Bruno Soares • um ano atrás

Obrigado! Jéssica
Mandou bem pacas...

^ | v • Responder • Compartilhar >



Leibnitz Fermat Sales Amorim • um ano atrás


muito obrigado pela clareza nesse tutor

^ | v • Responder • Compartilhar >



Robson Alves Moreira Pires • 2 anos atrás

Boa explicação. Sanou minhas dúvidas


^ | v • Responder • Compartilhar >



Ricardo Ianelli • 2 anos atrás

Opa, muito obrigado!

^ | v • Responder • Compartilhar >



anon0000001 • 3 anos atrás

muito bom!

^ | v • Responder • Compartilhar >

« [Choosing a Jekyll theme](https://jtemporal.com/choosing-a-jekyll-theme/) (https://jtemporal.com/choosing-a-jekyll-theme/)

[Python project dependencies: requirements.txt](https://jtemporal.com/requirements-txt-en/) » (https://jtemporal.com/requirements-txt-en/)



Recent articles

[Fechatag #40](https://jtemporal.com/fechatag-40/) (<https://jtemporal.com/fechatag-40/>) 30 Aug 2021

[Traveling Through a Secure API with Python and Auth0](https://jtemporal.com/traveling-through-a-secure-api-with-python-and-auth0/) (<https://jtemporal.com/traveling-through-a-secure-api-with-python-and-auth0/>) 24 Aug 2021

[Conheça o GitFichas!](https://jtemporal.com/conheca-o-gitfichas/) (<https://jtemporal.com/conheca-o-gitfichas/>) 21 Aug 2021

All rights reserved by [jtemporal](https://jtemporal.com) (<https://jtemporal.com>)

Built by [webjeda](http://webjeda.com) (<http://webjeda.com>)

 (<https://twitter.com/jesstemporal>)  (<http://github.com/jtemporal>) 
(<mailto:hello@jtemporal.com>)